November 21, 2022

1 Task 1

Training performance plot is shown in figure 1.

Figure 1: Training performance plot of task 1.

1.1 Question 1.1

Q: Discuss the effect of changing number of samples. How can this affect the performance and running time?

A: Increasing the number of samples lets us explore the action space much better, thus possibly increasing performance if a better (valued) action is found. However, running time also increases.

1.2 Question 1.1

Q: Assume that the dynamic model is learned from data during training, compare this method (CEM with a learned dynamics model) with a model-free RL algorithm, such as DDPG. Please list two advantages and two disadvantages of this method. (Hint: think about a real-time application such as controlling humaniod robot)

A: Advantage of CEM with learned dynamics model is that it allows further analysis, for example how would the system state behave with some unexpected input/action. Second advantage is that we can use the learned model in simulations for different RL algorithms. Disadvantage is that for some tasks model-free RL algorithm can produce sufficient results with lesser training (as it does not have to explicitly learn the model). Another disadvantage is that optimal-control kind of procedure can be too much for real-time applications, where as simple state-to-action run of model-free RL-algorithm can be doable.

1.3 Question 2.1

Q: Let's first think about the difficulty of the deep-sea task. 1) What is the probability of reaching the goal state (a function of N)? 2) If N is large, DQN (with the -greedy policy) usually fail to reach the goal state (in fact, N=10 is already challenging for DQN). In this case, which strategy will DQN converge to?

A: 1) $(1/2)^N$ with random guessing. 2) Because with N timesteps, there is a $(1/(2 * epsilon))^N$ chance of discovering the optimal path with an episode, which is very low if N is large.

1.4 Question 2.2

Q: Describe different phases in MCTS.

A: MCTS iteration consists of Selection, Expansion, Simulation, and Backup. Selection phase selects action according to the search policy, for example, puct. Expansion moves to the new node according to the selected action. Simulation simulates new value from the new node. Backup updates root node value based on the new node.

1.5 Question 2.3

Q: Explain how using Actor-Critic is beneficial in AlphaZero. Compare this to using Monte Carlo estimate of return function (Method you see on lecture slides).

A: Actor-critic allows learner to focus on temporal difference, which is beneficial as the monte-carlo estimate would require huge amounts of trajectories to reach the goal.

2 Task 2

2.1 Task 2.1 - 2.4

Implementation can be found in file az.py. The training performance of the implemented AlphaZero can be found in figure 2 with PUCT search policy and in figure 3 with greedy policy. As we can see, PUCT search performs much better than greedy. This is because PUCT can focus on different things than just the value, for example visit counts.

Figure 2: Training performance plot of AlphaZero with PUCT search policy.

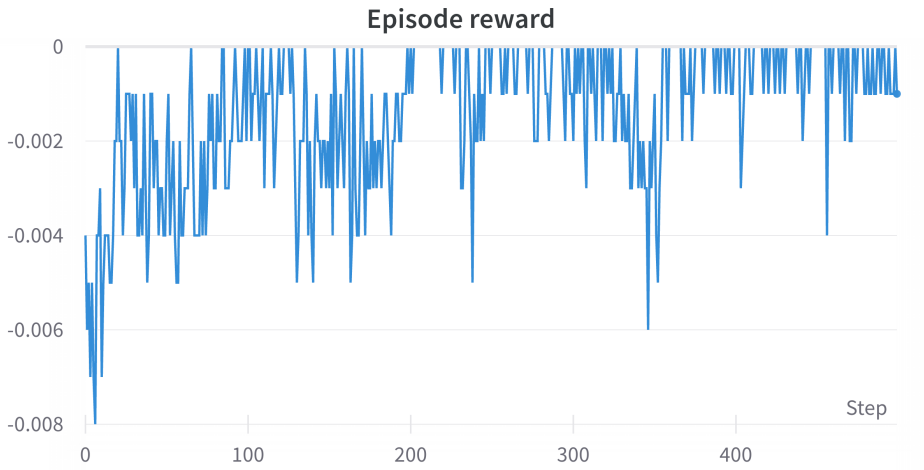

Figure 3: Training performance plot of AlphaZero with greedy policy.